What is claimed is:

1. A computer implemented method for encrypting data comprising the steps of:

creating an object key comprising data and methods that operate on said data; and

encrypting input plaintext data utilizing said object key in conjunction with an encryption process.

2. A computer implemented method as defined in Claim 1, wherein the encryption process comprises a block cipher system such that blocks of data bits are encrypted.

3. A computer implemented method as defined in Claim 2, wherein the object key is dynamic and changes with each data block encrypted.

4. A computer implemented method as defined in Claim 1, wherein the object key is dynamic.

5. A computer implemented method as defined in Claim 1, further comprising prior to the step of encrypting, the steps of:

creating an initial state of the object by the user;

creating an initial state of a random session object key;

encrypting the initial state of the random session object key in a block cipher encryption process with the initial state of the object key; and

modifying the object key based on seeding from the random session object key before each input data block so that each block is encrypted based on a different object key.

6. A computer implemented method as defined in Claim 3, wherein the initial state of the random session object key is created by generating a random number.

7. A computer implemented method as defined in Claim 6, wherein a new random number is generated and assigned as the initial state of the random session object key each time the block cipher encryption process is executed.

8. A computer implemented method as defined in Claim 5, wherein each object key is associated with a different key schedule for encrypting each input data block with said different key schedule.

9. A computer implemented method as defined in Claim 4, wherein the method of modifying the dynamic object key comprises the steps of:

generating a random seed unsigned byte and bit wise exclusive or to an unsigned byte of a current state of the object key provided by an incremented index into the current state of the object key (I_BYTE_OBJECT_KEY);

performing an unsigned byte addition on the output byte of the previous operation (PREV_OUTPUT) with I_BYTE_OBJECT_KEY;

performing a 16-bit multiplication of PREV_OUTPUT and I_BYTE_OBJECT_KEY modulus 254 and add 2;

performing a 16-bit addition of PREV_OUTPUT and I_BYTE_OBJECT_KEY;

performing another 16-bit addition of PREV_OUTPUT and I_BYTE_OBJECT_KEY;

performing a bit-wise exclusive or of PREV_OUTPUT with a 16-bit unsigned integer of the current state of the object key provided by an incremented index into the current state of the object key (I_INT_OBJECT_KEY);

rotating PREV_OUTPUT to the right I_BYTE_OBJECT_KEY modulus 15 plus 1 times;

performing a 16-bit addition of PREV_OUTPUT and I_INT_OBJECT_KEY;

performing a 16-bit multiplication of PREV_OUTPUT and I_INT_OBJECT_KEY with the lower order byte of I_INT_OBJECT_KEY modulus 254 plus 2;

performing a 16-bit addition of PREV_OUTPUT and I_INT_OBJECT_KEY;

performing another 16-bit addition of PREV_OUTPUT and I_INT_OBJECT_KEY;

performing a bit-wise exclusive or of PREV_OUTPUT with a 32-bit unsigned long integer of the current state of the object key provided by an incremented index into the current state of the object key (I_LONG_INT_OBJECT_KEY);

rotating PREV_OUTPUT to the left I_BYTE_OBJECT_KEY modulus 31 plus 1 times;

performing a bit-wise exclusive or of PREV_OUTPUT with I_LONG_INT_OBJECT_KEY;

repeating the previous set of operations eighty-four times substituting the random seed unsigned byte with a byte from a four byte output block provided by the previous set of operations recursively setting the current output block to a next output block when the current output block is exhausted, utilizing a different ordered byte each round;

performing a byte transposition of the bytes in the new 256 byte output block (N_OUTPUT) provided by the previous set of operations utilizing the following steps:

performing a byte-wise index through N_OUTPUT; switching the current byte of N_OUTPUT with the N_OUTPUT byte indexed at position I_BYTE_OBJECT_KEY; and indexing through the entire block of N_OUTPUT.

8.

10. A computer implemented method as defined in Claim 1, wherein said object key is dynamic and a modification method of said object key includes a hashing function.

9.

11. A computer implemented method as defined in Claim 1, wherein the object key is dynamic and includes at least two sub-object keys, and further wherein each sub-object key has a unique modification method associated therewith.

10.

12. A computer implemented method as defined in Claim 3, wherein the object key includes at least two sub-object keys and the random session object key operations with the object key are performed with only one of said sub-object keys.

11.

13. A computer implemented method as defined in Claim 3, wherein creating the initial state of the random session key object comprises the steps of:

accessing a running time clock in a computer;

multiplying together unique byte elements of the object key and summing and performing a bit-wise exclusive or to the time clock;

using the output of the previous step as a seed for a rand() function available in C libraries;

using an output of the rand() function modulus 255 plus an offset of 1 plus the lower eight bits of a high resolution computer clock timer is calculated and stored as one byte of the initial state of the random session object key;

repeating the previous set steps for each byte in the initial state of the random session key object.

12. 14. A computer implemented method as defined in Claim 3 8, wherein the modification method for the random session object key comprises the steps of:

indexing through each byte of the current state of the random session key object (I_BYTE_R_OBJECT) and replacing that byte with the output of the following operation:

double indexing into the object key with I_BYTE_R_OBJECT as a starting index and add an offset and I_BYTE_OBJECT_KEY.

Put A1

15. A computer implemented method as defined in Claim 17, wherein the modification method of said random session object key includes a hashing function.

14.
16. A computer implemented method as defined in Claim 3, wherein said object key is first initialized with the random session key object by using an initial current state of the random session key object to provide a key schedule in the modification method of the object key.

15.
17. A computer implemented method as defined in Claim 2, wherein input plaintext is compressed using a redundant byte reducing method and padded with random bytes to produce a file with a length that is evenly divisible by the block length so that the plaintext blocks are processed by said block cipher system.

16.
3
18. A computer implemented method as defined in Claim 1, further including the step of performing a keyed transposition of ciphertext bytes after all input blocks are encrypted.

17.
19. A computer implemented method as defined in Claim 2, wherein the encrypting step comprises the steps of:

transposition a substitution array whose elements contain unique numbers in reference to substitution array by switching a position of each element with a position provided by an element of a key.

18.
17
20. A computer implemented method as defined in Claim 19, wherein the position provided by an element of the key is bounded by the size of said substitution array.

21. A computer implemented method as defined by Claim 2, wherein the block cipher encryption process comprises the steps of:

transpositioning a substitution array whose elements contain unique numbers in reference to said substitution array by switching a position of each element with a position provided by an element of a key, which position provided by an element of the key is bounded by the size of said substitution array which is composed of 256 elements;

transpositioning a traverse array whose elements contain unique numbers in reference to said transverse array by switching a position of each element with a position provided by an element of the key, the position provided by an element of the key is bounded by the size of the transverse array which is equal to the block size;

replacing each input byte transverse number of times with the value of the substitution array indexed with the input byte;

summing each output byte of the previous three steps to an element of the key to create ciphertext;

grouping the ciphertext in a 32-bit sliding window and rotating to the left an element of the key modulus 31 plus 1 times, the window sliding by one byte after each rotation and this step being performed on all ciphertext bytes;

performing a bit-wise exclusive or of each cipher text byte to an element of the key;

transpositioning the substitution arry by switching a position of each element with a position provided by an element of the key, the position provided by an element of the key is bounded by a size of said substitution array;

transpositioning the transverse array by switching a position of each element with a position provided by an element of the key, the position provided by an element of the key is bounded by a size of said transverse array;

replacing each input byte transverse number of time with a value of the substitution array indexed with an input byte;

transpositioning the ciphertext by switching a position of each ciphertext element with a position provided by an element of the key, the position provided by an element of the key is bounded by a size of the block;

repeating the previous seven steps four times with the key elements being unique each time the key is accessed;

transpositioning each bit in the ciphertext block by switching a position of each ciphertext bit with a position provided by elements of the key, the position provided by elements of the key are bounded by the size of the blocks times eight; and

repeating the previous nine steps four times with the key elements being unique each time the key is accessed.

22. A computer implemented method as defined in Claim 21, wherein said key is the object key.

23. A computer implemented method as defined in Claim 21, wherein the last transpositioning step uses a switch key comprised of the following steps:

initializing the switch key with elements of an initial state of the object key;

grouping the switch key by 32-bit blocks;

replacing the current switch key element with the following process:

performing a bit-wise exclusive or of the current switch key element to a switch key element indixed two elements from the current element;

rotating the output of the previous step to the right switch key indexed three elements from the current element modulus thirty-one plus one;

performing a bit-wise exclusive or of the output from the previous step to a switch key element indexed three elements from the current element;

repeating the previous three steps for each final transposition switch operation.

22. 24. A computer implemented method as defined in Claim 21 23, wherein a hashing function is included in the creation of the switch key.

25. A computer implemented method for authenticating ciphertext, comprising the steps of:

generating a digital signature of a user using the ciphertext as input into a keyed one-way hash function; and

appending the digital signature to the input ciphertext.

26. A computer implemented method as defined in Claim 25, further comprising the steps of:

executing verification software which includes a user's secret digital signature key to regenerate the digital signature of the ciphertext; and

comparing the regenerated signature with the original signature appended to the ciphertext to determine whether the signatures are the same.

27. A computer implemented method as defined in Claim 25, wherein the step of generating includes the steps of:

dividing the input data into 256 byte data blocks;

further dividing the data block into 64 32-bit blocks (VAR_BLOCK);

modifying each VAR_BLOCK and element of a control key by a plurality of unique one-way irreversible hash functions;

repeating the previous step for all VAR_BLOCK - hash function combinations to create a running message digest;

modifying the running message digest using a bit-wise exclusive or to the next input data block;

repeating the previous four steps for all data blocks; and

transpositioning each byte of an output of the previous step by switching the position of each byte with another byte at a position provided by an element of the control key.

28. A computer implemented method as defined in Claim 25, wherein said keyed one-way hash function utilizes an object key.

29. A computer implemented method as defined in Claim 27, wherein the object key utilizes the output rounds of the one-way hash function to seed the object key's modification methods.

30. A cryptographic communications system comprising:

at least two networked computer systems linked by a communication channel; and

each computer system including a central processing unit and a memory storage device for executing a block cipher encryption/decryption process;

wherein the encryption process transforms an input plaintext message to a ciphertext message and the decryption process transforms the ciphertext message to the input plaintext message, the encryption/decryption process using a dynamic object key which changes with each block of input data, each object key being associated with a different key schedule to encrypt/decrypt the input plaintext/output ciphertext message.

24. 23
~~31~~. A cryptographic communications system as defined in Claim ~~30~~, wherein the encryption/decryption process further includes the use of a random session object key having an initial state randomly generated by the computer system, and wherein the object key modifications are based on seeding from the random session object key.

25. 24
~~32~~. A cryptographic communications system as defined in Claim ~~31~~, wherein an initial state of the object key is created by the user and wherein the initial state of the random session object key is created by the computer system generating a random number.

33. A cryptographic communications system as defined in Claim 30, wherein a digital signature is generated and appended to the ciphertext associated with each input plaintext message using a keyed one-way hash function utilizing an object key.

34. A cryptographic communications system as defined in Claim 30, wherein the block cipher encryption/decryption process includes use of a keyed transposition of a sequence of integers provides a ~~coat~~ count of substitution rounds for a particular input entering an S-box.

35. A cryptographic communications system as defined in Claim 30, wherein a digital signature is generated for each ciphertext file by using the ciphertext as input into the digital signature generation process.

33